Clouds

Clouds can be pretty. But clouds can also hide things, things it might sometimes be better to know about.

Images of clouds have been used when discussing networks for quite some time. When traditional telecoms companies were selling point-to-point circuits a drawing of a cloud was sometimes used. The cloud symbol helped indicate the provider's domain of responsibility, effectively hid the internal complexity of the network and focussed on the end user.

This was all fine when the product offered was an end-to-end circuit. What went in one end was what came out the other end. Users were expected to be only concerned about their end and the other end and the quality of the circuit in between. So for a while it was only about price and effectively a standard Quality of Service - QoS.

Then along came the internet...

In the early days of the public web many people presumed that networks continued to work just as telephone networks did. They of course used a telephone circuit to connect to the internet. And indeed their model was indeed not that wrong. People went to a web-site and in a sense they had a connection to that site. They could browse around on that site either by navigating within a page of information or clicking on a link that took them to another page on the same site. Or they would click on a link that would take them to another site - another connection? - and the process would continue.

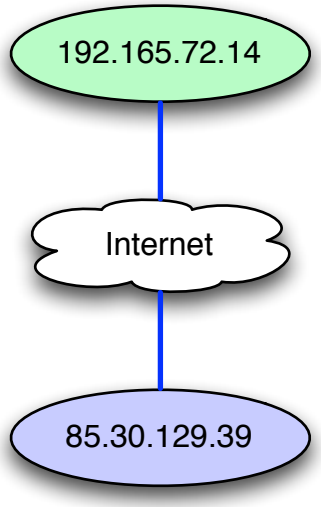

Figure 1: *Client (green) and web server (blue) using HTTP (blue lines)*

A lot of the recent discussion though about network neutrality and QoS seems to be predicated on that model. But is it really still good enough? Because the internet keeps changing. Or more accurately perhaps, people keep using it in different ways.

Now one of the great things about the internet is that the user can see some way into the cloud and this is what we did. We looked at what happened when a user went to a modern and not so atypical web-site. You can try the same thing and "your milage will vary!" But this is what we saw as we peered a little deeper and deeper into the cloud.

A user types “stupid.domain.name” into their browser.

Now for DNS queries most people’s computers off-load some of the work to a “resolver” - another, computer, a sort of proxy - on their local access network. For popular domain names the “resolver” will cache the response to previous queries. But given that there are a few hundred top-level domains and many millions of second and third-level domains the majority of domain names are not “popular”. So either the user’s computer or the resolver fires off in turn queries to a root server - pick 1 from 13 - and then to a TLD (top level domain) server - in this case pick one of the 10 name-servers for .NAME - and then...

In reality the resolution (as it is called) of a domain name to an IP address is a complicated process. In this case, if the name-server chosen for .NAME is c6.nstld.com, then the client must first resolve that name before it can resolve stupid.domain.name. And so it continues in a recursive process until all names involved are resolved to their respective IP addresses.

So there are interactions with about ten name-servers just to get the address associated with “stupid.domain.name” and the user’s computer is ready to do http://stupid.domain.name/

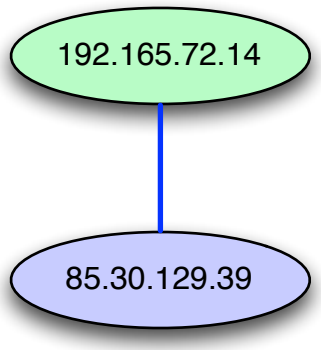

Figure 2: *Client connect to web server*

Are we finished?

Well not quite!

First of all a typical web-page consists of multiple objects - chunks of text and various images. So there are local links which result in more http requests to download, for example, the images. And of course there are the passive links to other web-pages of the same server or other servers. The user decides whether to click on those or not, whether to go to the other page or not.

But the home-page for “stupid.domain.name” also contains a number of active links to content on other servers. To completely and properly display the selected home-page these other servers, servers elsewhere and on other networks, also have to be contacted.

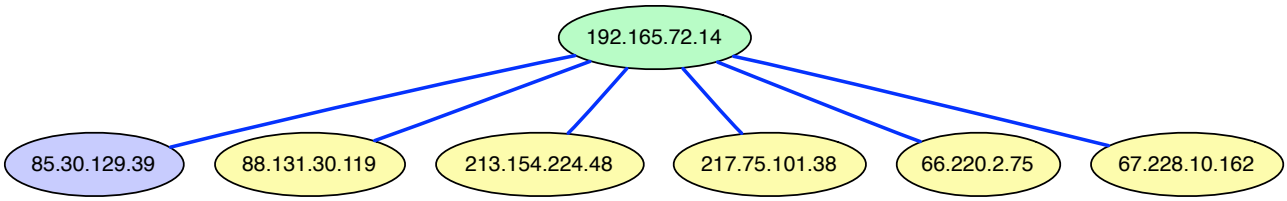

Figure 3: *Web server (blue) and additional content servers (yellow)*

So we have another 5 series of separate DNS resolutions, each resulting in multiple name-servers being contacted.

To conclude, to display just the home-page of “stupid.domain.name” the client have been in contact with 6 content-servers, and one name-server, which in turn have had interactions with 12 name-servers.

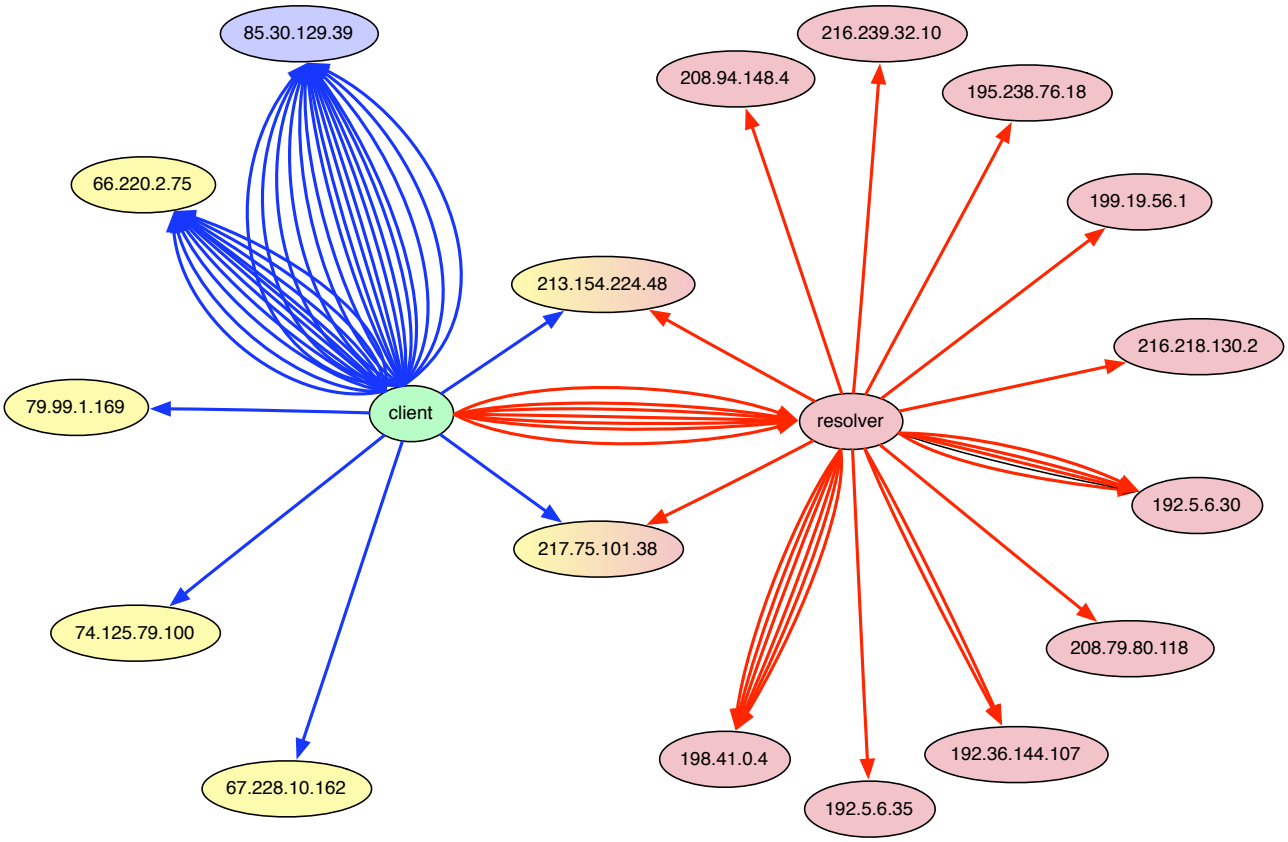

Figure 3: *All HTTP (blue) and DNS (red) requests for one web page*

This is illustrated in Figure 3 where we see name-servers as red, and content-servers as yellow. The lines indicate transactions either HTTP (blue) or DNS (red). We also see that two hosts act both as name-servers and as content-servers.

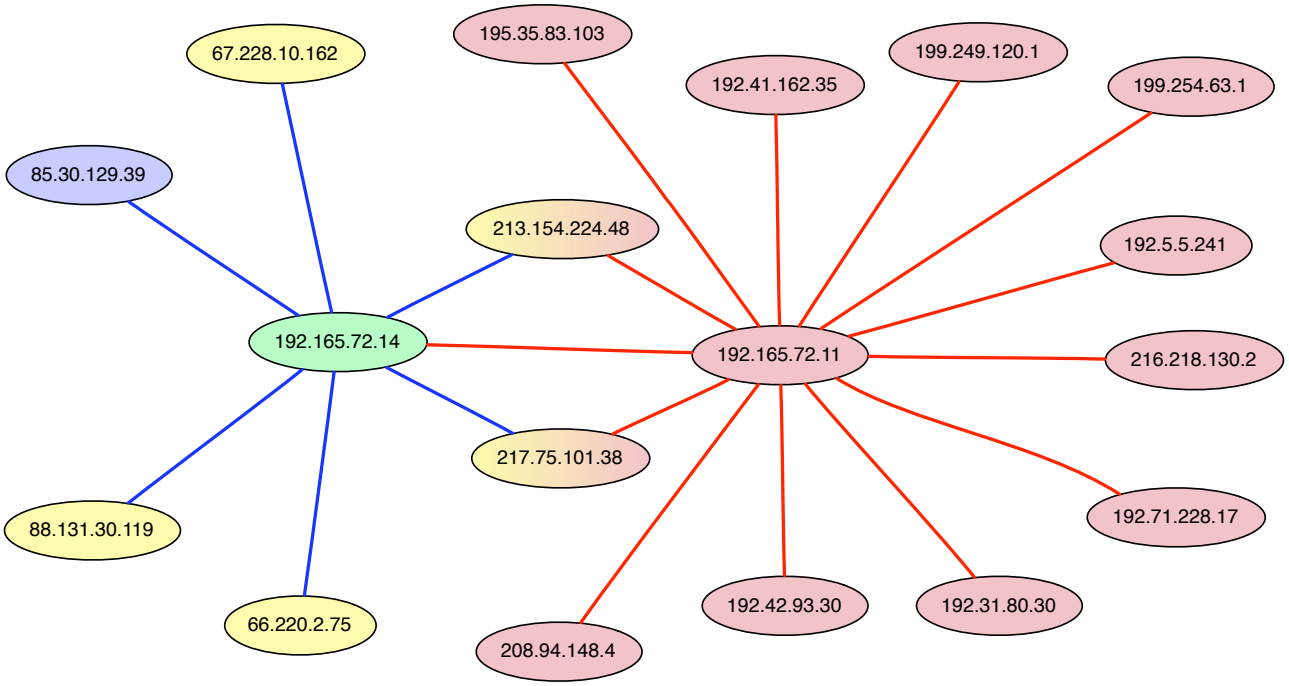

Figure 4: *Client (green), content servers (blue and yellow) and name-servers (red)*

Are we finished?

Well still not quite. We can still peer a bit further into the cloud and still see a bit more.

Where are the servers we have mentioned? Which networks are they on? And which intervening networks need to be traversed?

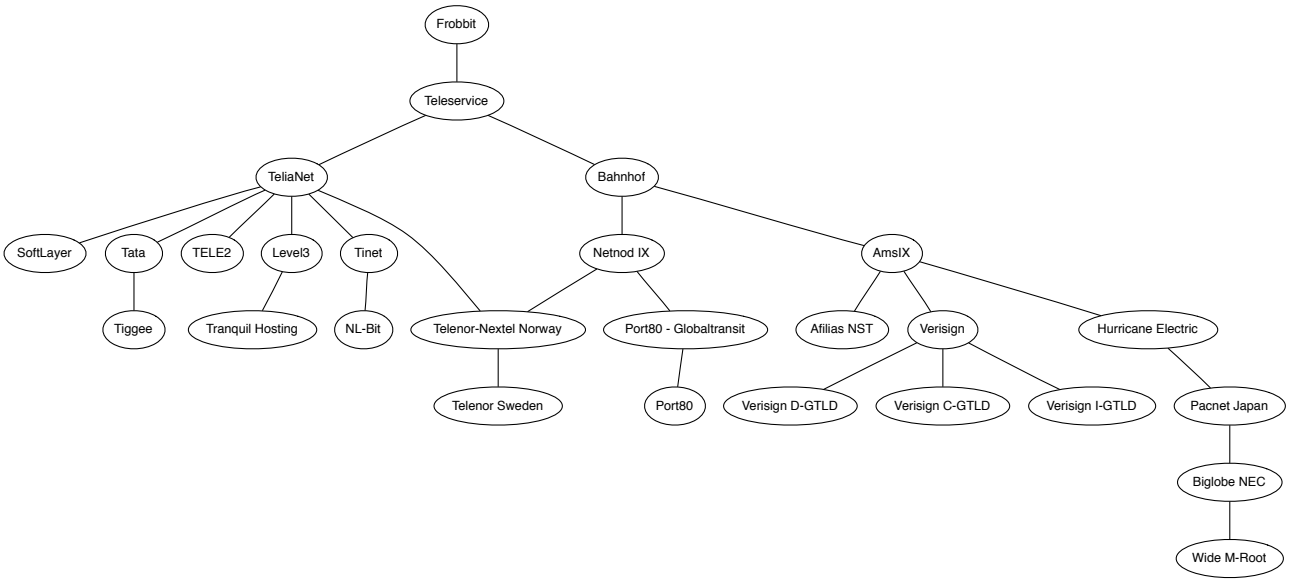

Figure 5: *All the network providers involved for this web page*

A bit of playing around and we identified about 27 autonomous networks which are involved either in hosting the servers or providing transit.

Are we finished?

Not necessarily. We could probe a little the activities of some of the boxes involved - but obviously not all. There are various “boxes in the middle” doing what “boxes in the middle” do. From routers to proxies, from NATs to firewalls, from load balancers to content caches, in this specific case more than 100 boxes involved! And then there are tunnels...

But we will pause here.

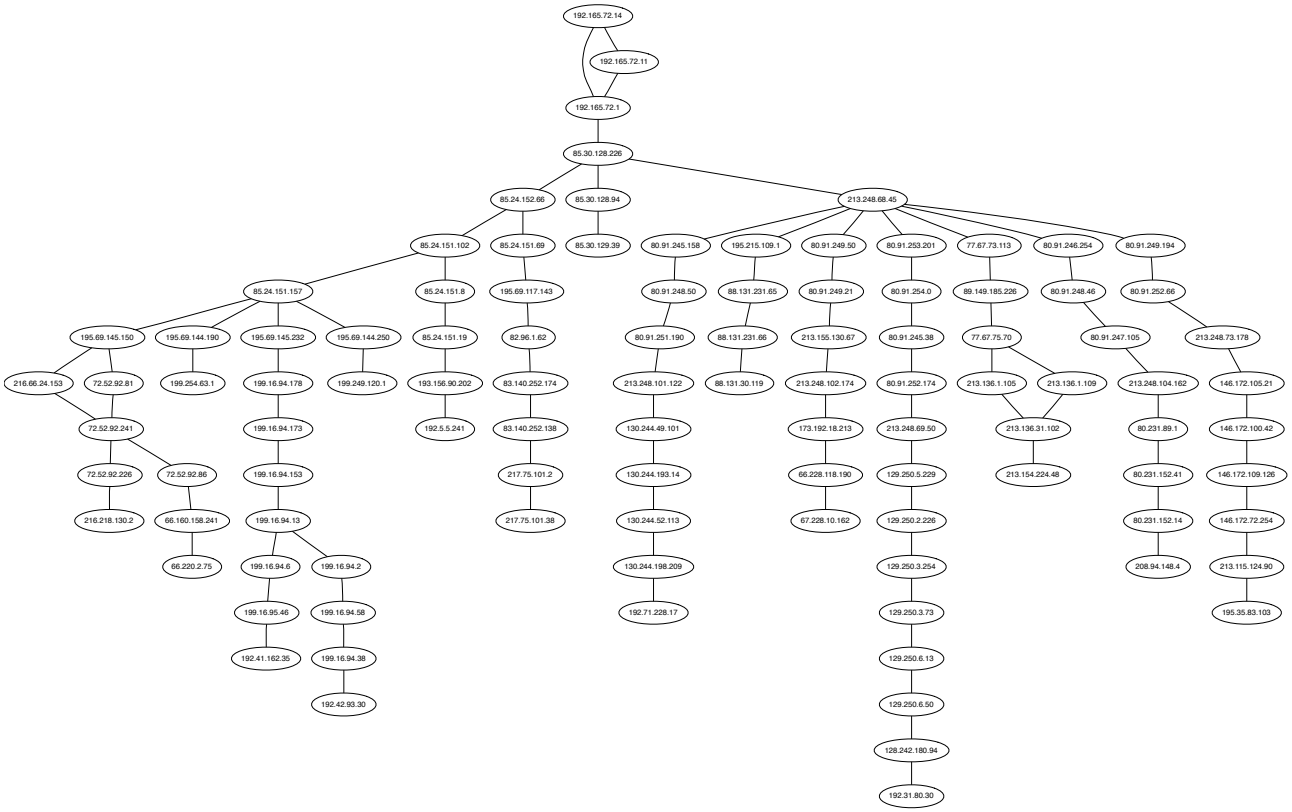

Figure 6: *Some of the network elements (hosts, routers etc) involved in the transaction*

So in this case a user looking at one home-page involves 17 servers (name servers plus content-servers) scattered across more than 27 networks (hosting and transit).

Of course each of these servers has only so much local access bandwidth and so much processing power. Indeed some content/service boxes will decide unilaterally how much of their resources they will devote to any incoming request. And to state the obvious: your access provider does not have any contract with the average content/service provider.

Anyway our typical user might then just decide that “stupid.domain.name” was not the site they should be looking at right now and decide go somewhere else. So the process would start again. There are potentially lots of places they could now go to though. .EU has well over 3 million domain names registered and .SE has another one million and so on. And again your access provider has contracts with almost none of them.

On the other hand the user might just go off and do something else: go for a beer or read a book. The web-page is of course “still there”. So the “circuits” are still active, still being used? Well no. There were and are simply no circuits in the traditional sense. The DNS in particular is very forgetful. A query comes in and a response is sent back. And the transaction is forgotten. The response may be cached for a while to help with responding to another query from another user. But that is it. The content on the screen? Well that has been delivered. Until the user or their browser requests more content or some content

needs to be refreshed there is no further interaction with the content server. So there is no need to shut down the browser before reaching for that beer and a book.

This glimpse into the cloud should hopefully have helped in understanding a bit of what is going on now and why traditional notions like circuits and some current suggestions about QoS are not quite what they seem. They just do not make sense anymore. You cannot buy a circuit with a defined QoS to any web-site of your choosing. You do buy connectivity into the internet - the cloud? - and that allows the mass of interactions that are needed even to just look at a simple web-page these days.

The image of a cloud remains useful. And even pretty. But we have to remember sometimes what is inside. And what may be inside.

:-)